(12) **UK Patent Application** (19) **GB** (11) **2 363 952** (13) **A**

(43) Date of A Publication 09.01.2002

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Nan Feng
Dong Liu
Leo Y Liu

(74) Agent and/or Address for Service
IBM United Kingdom Limited
Intellectual Property Department, Hursley Park,
WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(54) Abstract Title
Measuring response times to balance load among mirror servers

(57) A method for balancing load among a plurality of
mirror servers, wherein a user may select and get access
to any one of said plurality of mirror servers within an
identical web page, comprising steps of: (1) when said
web page is accessed by a client, transmitting not only
said web page but also a predetermined script to said
client; (2) automatically executing said script at said
client so as to respectively create connections with each
of said plurality of mirror servers and measure
respective response times; (3) selecting the mirror server
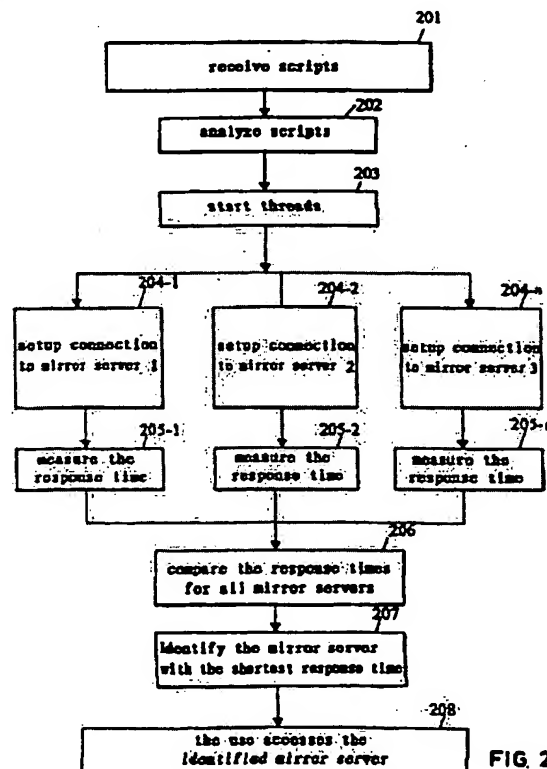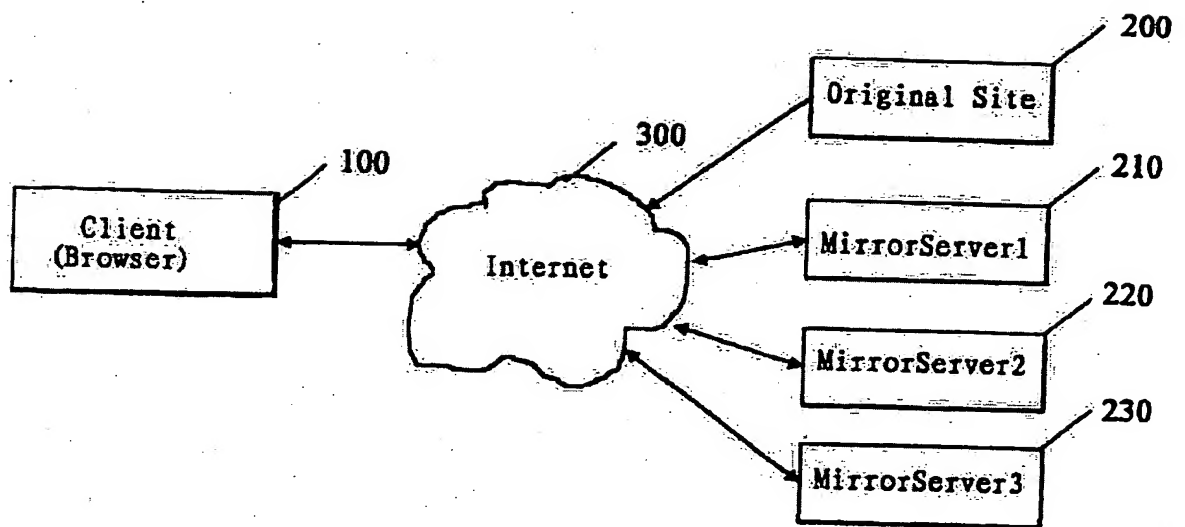having the shortest response time for users to access.

FIG. 2

GB 2 363 952 A

FIG. 1

201

```
receive scripts
```

202

```
analyze scripts
```

203

```
start threads
```

204-1

```
setup connection
to mirror server 1
```

204-2

```
setup connection
to mirror server 2
```

204-n

```
setup connection
to mirror server 3
```

205-1

```
measure the
response time
```

205-2

```
measure the
response time
```

205-n

```
measure the
response time
```

206

```
compare the response times
for all mirror servers
```

207

```
identify the mirror server
with the shortest response time
```

208

```
the use accesses the
identified mirror server
```
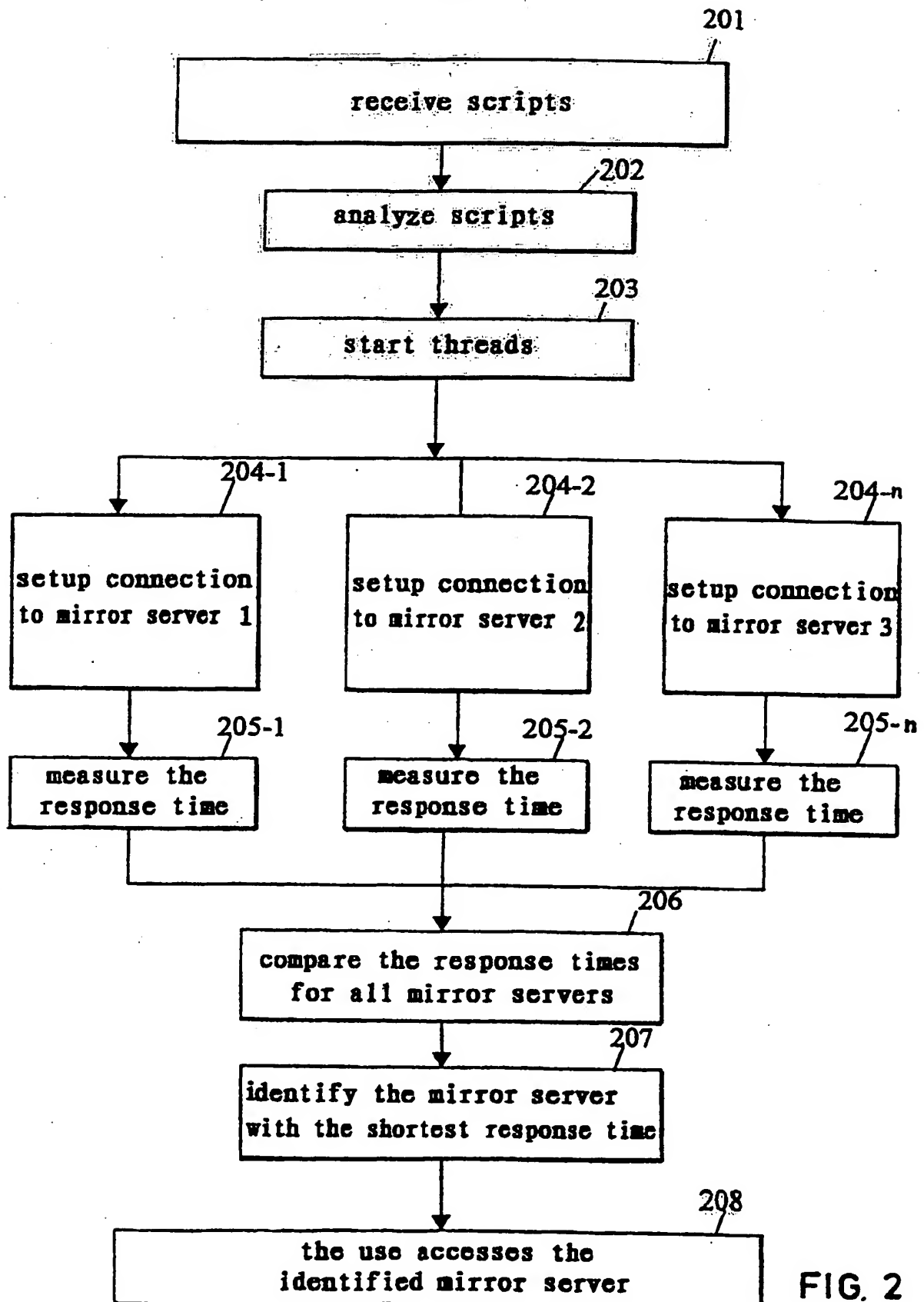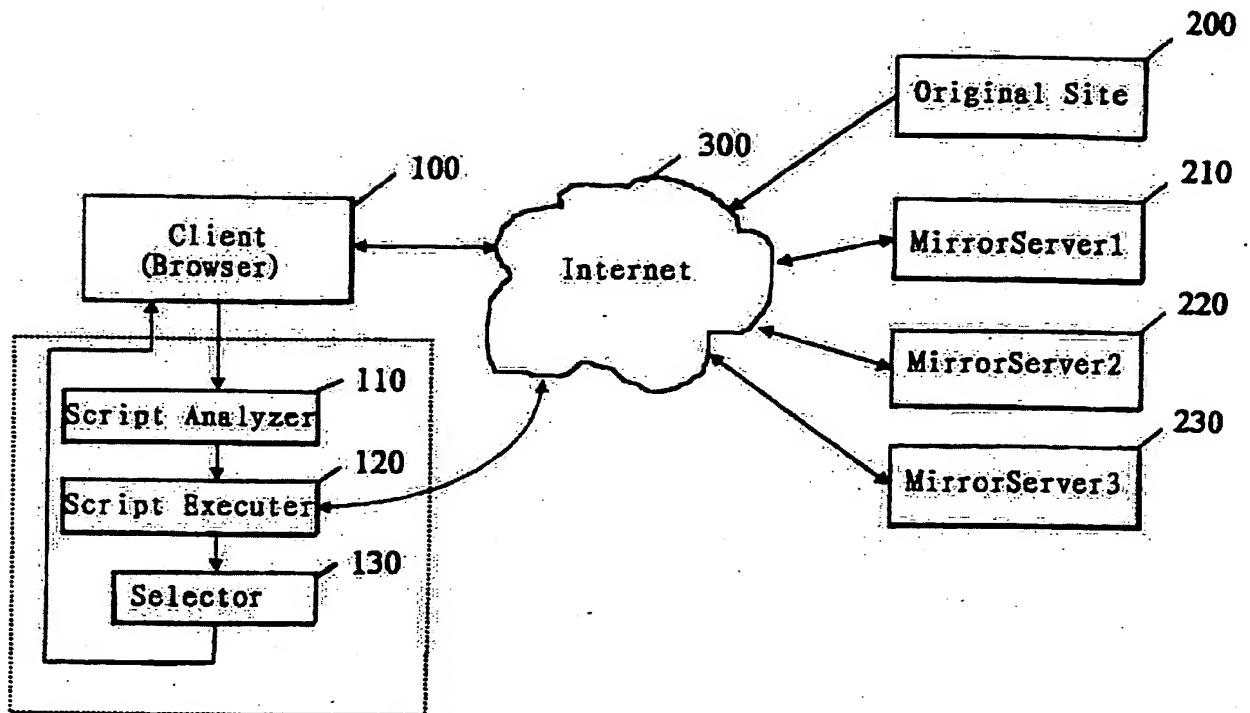
FIG. 2

FIG. 3

A METHOD OF BALANCING LOAD AMONG MIRROR SERVERS

The present invention relates to a method and apparatus that can balance load among a plurality of mirror servers, and more particularly to a method of balancing the servers load with the active participation of clients.

World Wide Web (WWW) is a multimedia information retrieval system on Internet. It is the most common way to transfer data over Internet. Some other means include FTP (File Transfer Protocol), Gopher and etc. On the web, clients can achieve transactions on servers by HTTP (Hypertext Transfer Protocol), and HTTP is a well-known application protocol. This protocol allows clients use standard HTML (Hypertext Markup Language) pages to access all kinds of files (Text, Image, Sound, Video and etc). HTML files provide the fundamental file format and enable developers define links, which link to other server sites. Under Internet circumstance, we can use URL (Uniform Resource Locator) to define a certain servers address or even the network path. URL has a special syntax to define the network path.

A typical URL is like this: http://www.your_company.com/path/, "your_company" is the host server name, "path" is the directory, in which page can be found. A Name Server can translate an URL into an IP address. Name Server on Internet is called DNS (Domain Name Server). The process that web clients ask DNS to translate host name to IP address is called resolution. In TCP/IP, Name Server will translate the Host Name into one or several IP address list. The IP list will be sent back to those clients who ask HTTP requests. Each IP address locates a server, this server will process the request sent by web client using a web browser.

WWW adopts HTML and follows Client/Server architecture. HTTP service client uses web browsers, which can send all kinds of requests to the server and display the HTML files (sent back from the server) on the screen.

With thousands of companies, universities, government organizations posting their own Home Page on Internet, Internet becomes a very precious information resource. Even a new user with only a little practices can visit millions pages and thousands new groups. Internet accesses and the related markets are developing fast too.

In order to provide a high performance service and support more concurrent users, some big companies setup several mirror server. All these servers are deployed in different regions or even different countries. Each

server has its unique network path (URL) but provides the same service functions.

But, the deployment of the server is always determined by experiences and cannot reflect the real access pattern. If the regions are not select wisely, overload costs will increase inevitably.

Even worse, most users choose a site from a list of mirror site randomly. The most common way is to select the nearest mirror. But, the complicate situations in network cannot make sure the nearest one is the fastest one.

For example, a user wanted to download certain software from Internet. He/She got a list of server sites. Each server in this list, such as www.download.com, www.microsoft.com and www.linux.org, could provide this software. In most cases, a user wants to select the fastest one, by which he could get what he want in the minimum time. Unfortunately, most users are not network specialists, or they don't have enough network tools. So most of them will select one of them randomly. Another possible situation is that some users will select the nearest site by location. They regard the nearest site should have the shortest response time. Unfortunately again, the network speed to some site determined by the workload of the server, the topology of the network and some other more important issues. As users cannot take the real load of a server into mind before their selection, different mirrors may have different work statuses. The workload is not well balanced among these servers. The worst, if a user selected a server with heavy load already but with the nearest location, he must spend a longer time to download the software he wanted.

Due to the inconsequence of deployment and blindfold selection, the load among mirror sites is not balanced. So the overall performances of the Internet are decreased. For reasons above, it is a very critical problem to balance the load among the mirrors.

As we know, the current load balance methods are only deal with the LAN, and all these methods are only work on the server side. In order to make the balance transparent to end-users, all these method must be devised carefully. Due to these limitations, all these method designed for LAN cannot be used on Internet directly and easily.

The first object of this invention is to provide a method, which can balance the load among mirrors with clients' active participation. The method just needs a few modifications on clients.

The second object of this invention is to provide an apparatus, which can balance the servers' load and this apparatus can be easily installed into clients.

In order to achieve the first object, this invention provides a method for balancing load among a plurality of mirror servers, wherein a user may select and get access to any one of said plurality of mirror servers within an identical web page, said method characterized by comprising steps of:

(1) when said web page is accessed by a client, transmitting not only said web page but also a predetermined script to said client;

(2) automatically executing said script at said client so as to respectively create connections with each of said plurality of mirror servers and measure respective response times;

(3) selecting a mirror server having the shortest response time for the users to access.

In order to achieve the second object, this invention provides an apparatus for balancing load among a plurality of mirror servers, characterized by being installed in a client machine and comprising: a script analyzer, for analyzing a predetermined script received by a client; a script executor, for respectively creating connections with each of said plurality of mirror servers and measuring respective response times based on analyzed result from said script analyzer; a selector, for selecting the mirror server having the shortest response time for users to access.

The method and apparatus of this invention can reflect the real circumstance more accurately with the help of clients active participation. Although, this method needs participation of clients, but all these actions could be done transparent to clients. The end-users will never aware it.

Another benefit of this invention is that this method can help the end-user find the fastest server from a list to accelerate the request-response cycle easily.

Besides, the apparatus of this invention can help mirror sites get the desired information from its clients automatically. This information may help network administrator to analyze clients access pattern. By counting the connections, administrators can control its mirror servers more effectively. For example, he may add some servers to that place where

many connections occur and remove some servers from the place where only a few connections occur.

By the figures, we will illustrate the embodiment of this invention
5   in details. The benefits and advantages of this invention will become more obvious.

FIGURE 1 is the running environment of this invention.

10   FIGURE 2 is the flowchart of the method, which can balance the load among popularity mirror servers.

FIGURE 3 depicts the framework of the apparatus, which may help the balance among popularity mirror servers.
15
Next, we will describe the preferred embodiment of this invention.

In Figure 1, a client 100 sets up a connection with a host server 200 through Internet 300. In this example, the client 100 might use any kinds
20   of browsers, such as Internet Explorer from Microsoft or Netscape Communicator. After the connection was setup, the Client 100 got a page and showed this page to the end-user. In this page, a lot of mirror sites' URL addresses were listed. In Figure 1, we drew three mirror servers. They are mirror server 210, mirror server 220 and mirror server 230. The user could
25   use some kinds of pointing device such as a mouse to choose a site he wanted to visit. Client 100 and each server complied with the Client/Server architecture, as we all know. Numeral 300 stands for Internet.

To implement this invention into Figure 1, we need some work on
30   client only. Before we implement the invention, we must build some small scripts and store them on the host 200. By doing some small tricks on client (installing a certain small software or download it from Internet automatically, this software can interpret the script we wrote on host just), the scripts can be run without the end-users' awareness after they
35   were downloaded together with the page. The method to build the script and the interpret engine is quite known in computer field.

In the script, we can define the URL address of each mirror site (URL), connection time-out setting, times of retry connection. As an
40   extension, we can write some feedback information in the script too. These feedbacks include the IP address of clients, domain names, operation platforms, browser types and etc.

When the host server 200 got the request from the client 100, the host server 200 would send the requested page together with the script we already had written. After getting the script, the client 100 would run the script automatically as well as displaying the page to the users.

The script interpret engine would setup connections to each mirror site according to the description in the script and measure the response time from the servers. Here, all the action may be done in a multi-thread mode. Then, by analyzing the response time of each server, a server with the shortest response time would be selected. We had many choices to show this result to users. For example, we could show a list with all servers and its response time on, and use some extraordinary drawing work on the fastest one to make it more outstanding. Of course, the display method does not belong to this invention.

We may implement the script interpreter as a plug-in card, and store them on the server site 200. The script can be restarted by users as well as starts by itself.

In Figure 2, we illustrate the working process of this invention in more details. In step 201, receiving scripts from the host server. In step 202, analyze the scripts and get the desired information. In step 203, start several threads according to the analyze results. These thread will finish step 204-1,204-2,...,204-n and 205-1,205-2,...,205-n. Here n stands for the number of mirrors. The interrupter will start one thread to one mirror.

Take the mirror 1 as an example. In step 204-1, setup a connection with the first mirror. Then, in step 205-1 it will measure the response time of mirror 1. In step 206, all response times from different servers are compared here, and select one with shortest response time. In step 207, show the fastest site to the end-users. Finally, in step 208, the end-users may select the fastest to send his subsequent requests.

By using this invention, the mirror with the shortest response time can be selected automatically. And this selection can be made while the user accessing the host 200 dynamically. That is to say, the site with shortest response time is determined dynamically too. Because in most cases users prefer the fastest site, mirror 210, mirror 220 and mirror 230s load are balanced with the active participation of the users.

Table 1 gives out a sample script, which can be processed in Figure 2. In this script, HTTP is the connection protocol. Three mirror servers are defined. Their URLs are www1.ibm.com, www2.ibm.com and www3.ibm.com.

Connection Time-Out is 100 seconds. During the measurement, it will test the target server three times and calculate the average time. This script also define that it will stop as long as it finds the fastest one. Users can restart the script. After the connections are setup, clients will send some information back to servers, such as IP address, Platform name and browser type.

TABLE 1

```
#This is a sample script
#Define connection type
CONNECT=HTTP
#List all mirrors
SITES=www1.ibm.com, www2.ibm.com, www3.ibm.com
#Define the Time-Out
TIMEOUT=100
#Test each mirror three times.
TEST_COUNT=3
#Stop the script as long as find the fastest one
DISPLAY_COUNT=1
#Users can restart the script by hand
CAN_RESTART=TRUE
#Feedbacks to server
FEEDBACK=IPADDRESS, PLATFORM, BROWSERTYPE
```

Figure 3 depicts how to use an apparatus to balance the load among mirror servers. We can install such apparatus into clients just like the dot-line shown in Figure 3. The client 100 can execute the apparatus and get results from the apparatus. This apparatus includes a script analyzer 110, a script interpreter 120 and a selector 130. After clients get scripts from the original host server 200, the script will be sent to the script analyzer 110. The script analyzer 100 will analyze this script and get the information defined in it, such as URLs for each mirror site, connection time-out setting, times of retries, clients feedbacks and etc. The script interpreter 120 will setup connections and measure each server's response time according to the analyzed results from the script analyzer 110. The selector 130 will choose the mirror with the shortest response time for users next action.

Although we have discussed the preferred embodiment of this invention in conjunction with the drawings, it is still very easy for those skilled in the art to make some modifications without departing from the spirit of this invention. So, the scope of this invention is just limited by the claims.

CLAIMS

1.   A method for balancing load among a plurality of mirror servers, wherein a user may select and get access to any one of said plurality of mirror servers within an identical web page, said method characterized by comprising steps of:

(1) when said web page is accessed by a client, transmitting not only said web page but also a predetermined script to said client;

(2) automatically executing said script at said client so as to respectively create connections with each of said plurality of mirror servers and measure respective response times;

(3) selecting a mirror server having the shortest response time for the users to access.

2.   The method according to claim 1, characterized in that in said step (1) said predetermined script is transmitted together with said web page to said client.

3.   The method according to claim 1 or 2, characterized in that said step (2) comprises steps of :

calling a predetermined engine by said client;

executing said script by said engine, so as to respectively create connections with each of said plurality of mirror servers and measure respective response times.

4.   The method according to claim 1, 2 or 3, characterized in that said step (2) is performed in a multi-thread manner for said plurality of mirror servers.

5.   The method according to any one of claims 1 to 4, characterized in that in said step (2) the client information is also sent to the mirror servers being connected.

6.   The method according to claim 5, characterized in that said client information includes at least one of IP address, domain name, platform name, platform version, and browser type of said client.

7.   The method according to any one of claims 1 to 6, characterized in that said connections in said step (2) are created through the proxies.

8.   The method according to any one of claims 1 to 7, characterized in that said script can be re-started by said user.

9.   The method according to any one of claims 1 to 8, characterized in that said step (3) further comprises a step of:

comparing respective response times of said plurality of mirror servers.

10.   The method according to claim 9, characterized in that said step (3) further comprises steps of:

notifying said user of the mirror server having the shortest response time;

getting access by the user to the mirror server of which said user has been notified.

11.   An apparatus for balancing load among a plurality of mirror servers, characterized by being installed in a client machine and comprising:

a script analyzer, for analyzing a predetermined script received by a client;

a script executor, for respectively creating connections with each of said plurality of mirror servers and measuring respective response times based on analyzed result from said script analyzer;

a selector, for selecting the mirror server having the shortest response time for users to access.

12.   The apparatus according to claim 11, characterized in that said script is transmitted together with said web page to said client.

13.   The apparatus according to claim 11 or 12, characterized in that said script executor operates in a multi-thread manner for said plurality of mirror servers.

14.   The apparatus according to claim 11, 12 or 13, characterized in that said script executor also sends the client information to the mirror servers being connected.

15.   The apparatus according to claim 14, characterized in that said client information includes at least one of IP address, domain name, platform name, platform version, and browser type of said client.

5   16.   The apparatus according to any one of claims 11 to 15, characterized in that said script executor makes connections with mirror servers through the proxies.

17.   The apparatus according to any one of claims 11 to 16, characterized
10   in that said script executor can be re-started by said user so as to execute said script.

18.   The apparatus according to any one of claims 11 to 17, characterized in that said selector comprises a comparator for comparing respective
15   response times of said plurality of mirror servers.

19.   The apparatus according to claim 18, characterized in that said selector further comprises:

20   means for notifying said user of the mirror server having the shortest response time;

means for receiving selection made by a user on the mirror servers.

25

INVESTOR IN PEOPLE

| Application No: | GB 0101768.0 | Examiner: | Stephen Brown |
| Claims searched: | 1-19 | Date of search: | 16 October 2001 |

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H4P (PPEC), G4A (FN).

Int Cl (Ed.7): H04L: 12/56, 29/06, 29/12.

Other: Online: WPI, EPODOC, JAPIO.

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP 0 959 601 A1 (Sun) See especially the abstract, figure 4, and column 4, lines 14 to 35.	-
A, P	WO 00/14633 A1 (Sun) See especially the abstract	-
A, P	WO 00/10084 A2 (Microsoft) See especially the abstract, figure 2, and page 8, line 28, to page 9, line 20.	-
A	US 5 459 837 (Digital Equipment) See especially the abstract and page 2, lines 12 to 36, and page 5, lines 3 to 46.	-

An Executive Agency of the Department of Trade and Industry